REMARKS

Claims 1-22 and 27-30 are currently pending. Claims 23-26 and 31-33 have been canceled. Claims 1, 20, 27, 28 and 30 have been amended. Favorable reconsideration of the office action mailed October 23, 2006, is requested in view of the foregoing amendments and the following remarks.

35 U.S.C. § 101

Claims 20-30 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Claims 20 and 27 have been amended. Withdrawal of the 35 U.S.C. § 101 rejections of claims 20-22 and 27-30 is requested.

Claims 23-26 have been canceled thereby rendering the rejection moot.

35 U.S.C. § 102

Claims 1-22 and 27-30 were rejected under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent No. 5,854,932 ("Mariani").

Claim 1. The examiner rejected claim 1, stating that

> Mariani anticipates a method comprising: providing a system including an interface (Col 1, lines 39 - 46) and multiple units of compiled code (Col 1, lines 39-52), the interface including global components (Col 13, scope, lines 25 - 45) and each unit depending on at least one of the global components included in the interface (combination if limitations above); dividing the interface into levels ((col 13, levels of dependency shape, lines 25 - 35), each level including a set of one or more of the global components (Highest level must exist - at least one level); ...

It is not clear to the applicant where in the cited portions of Mariani the examiner finds the alleged teaching of "dividing the interface into levels, each level including a set of one or more of the global components," as recited in previously-presented claim 1. In col. 13, lines 25-35, Mariani states:

> ... the dependency types of an object code file on a class that are tracked by the minimal rebuild system 100 include dependencies on a search for a name in the scope of the class, a dependency on the shape of the class's data, and a dependency on the shape of the class's vtable. The set of these dependency types for each class is represented in an instance (hereafter the object code file's "dependencies data structure" on the class) of a data structure 120 which comprises a hash array 122, and flags 124-125. (Instances of the data structure 120 also are utilized by the minimal rebuild system 100 in tracking changes to classes as described below.)

In col. 14, line 64 – col. 15, line 16, Mariani further states:

> The minimal rebuild system 100 sets the vtable shape dependency flag 125 to represent a dependency of the object code file 82 on the shape of the class's vtable. *A vtable is an array containing pointers to locations of the class's virtual function members, and conventionally utilized as an interface for program code outside the class to access those members.* The vtable has a shape determined by the number of pointers and their offsets (i.e., position) within the vtable. Code outside the class (e.g., in the object code files 82) accesses the class's members according to the offsets of their pointers within the vtable. Accordingly, an object code file which calls a class's member through its vtable generally has a dependency on the class's vtable. An object code file's dependency on the shape of a class's vtable is detected by the compiler 80 while compiling a compiland into the object code file that contains any call to a virtual function of the class. When the compiler 80 detects this vtable shape dependency, it notifies the minimal rebuild engine 102. In response, the minimal rebuild engine 102 sets the vtable shape dependency flag in the data structure 120 that represents the object code file's dependencies on the class. (emphasis added)

Even if a vtable of Mariani corresponds to an "interface" of claim 1 as the examiner appears to suggest, no portion of Mariani, cited or otherwise, teaches or suggests "dividing the interface into levels, each level including a set of one or more of the global components, each global component being included in no more than one of the levels," as recited in amended claim 1.

Further, claim 1 recites "generating multiple dependency lists; associating a unique one of the multiple dependency lists with each of the levels; associating a unit with a dependency list based on the global components on which the unit depends; and marking only those units

associated with a particular dependency list for recompilation based on a change to a particular global component affecting those dependency lists with relationships to a level that includes the changed global component."

Nothing in Mariani teaches or suggests an association between a dependency list and a unit of compiled code, or an association between a dependency list and a level of an interface, where the level includes a set of one or more global components. At most, Mariani discloses an association between an object code file and a class.

For at least the foregoing reasons, claim 1 should be allowed.

Claims 20 and 27 contain limitations similar to the limitations of claim 1 and should be allowed for at least the same reasons.

All of the dependent claims are allowable for at least the same reasons set forth with respect to the claims from which they depend.

Conclusion

For the foregoing reasons, the applicant submits that all the claims are in condition for allowance.

By responding in the foregoing remarks only to particular positions taken by the examiner, the Applicants do not acquiesce with other positions that have not been explicitly addressed. In addition, the Applicants' arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 1/10/07

Mandy Jubang
Reg. No. 45,884

Customer No. 32864
Telephone: (617) 542-5070
Facsimile: (617) 542-8906